

Method of updating information in a translation lookaside buffer.

Publication number: DE3689276 (T2)

Publication date: 1994-02-24

Inventor(s): BRYG WILLIAM R [US]; GARDNER MICHAEL E [US];
BOETTNER STEVEN C [US]

Applicant(s): HEWLETT PACKARD CO [US]

Classification:

- International: G06F12/10; G06F12/10; (IPC1-7): G06F12/10; G06F12/02

- European: G06F12/10L4

Application number: DE19863689276T 19860620

Priority number(s): US19850750390 19850628

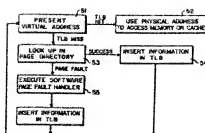
Also published as:

EP0212129 (A2)
 EP0212129 (A3)
 EP0212129 (B1)
 JP62003357 (A)
 CA1262287 (A1)

Abstract not available for DE 3689276 (T2)

Abstract of corresponding document: EP 0212129 (A2)

Explicit instructions are provided that enable software to directly control insertion of information into a translation lookaside buffer (TLB). A first pair of instructions enable information to be inserted into a data TLB and a second pair of instructions enable information to be inserted into an instruction TLB. In each of these pairs, the first instruction inserts the virtual address and the associated physical address. In response to the second instruction of each of these pairs, additional information about that physical page, such as protection information and flags, is inserted.



Data supplied from the esp@cenet database — Worldwide



19 **BUNDESREPUBLIK
DEUTSCHLAND**



**DEUTSCHES
PATENTAMT**

12 **Übersetzung der
europäischen Patentschrift**

27 **EP 0 212 129 B1**

10 **DE 36 89 276 T 2**

51 Int. Cl.⁵:
G 06 F 12/10
G 06 F 12/02

- | | | |
|----|---|--------------|
| 21 | Deutsches Aktenzeichen: | 36 89 276.9 |
| 26 | Europäisches Aktenzeichen: | 86 108 415.0 |
| 36 | Europäischer Anmeldetag: | 20. 6. 86 |
| 37 | Erstveröffentlichung durch das EPA: | 4. 3. 87 |
| 37 | Veröffentlichungstag
der Patenterteilung beim EPA: | 10. 11. 93 |
| 47 | Veröffentlichungstag im Patentblatt: | 24. 2. 94 |

DE 36 89 276 T 2

30 Unionspriorität: 29 33 31

28.06.85 US 750390

73 Patentinhaber:

Hewlett-Packard Co., Palo Alto, Calif., US

74 Vertreter:

Kohler, R., Dipl.-Phys.; Schmid, B., Dipl.-Ing.;
Holzmüller, R., Dipl.-Ing. Dr.-Ing.; Bähring, A.,
Dipl.-Phys. Univ. Dr.rer.nat.; Rüdel, D., Dipl.-Ing.
Dipl.-Wirtsch.-Ing., Pat.-Anwälte, 70565 Stuttgart

84 Benannte Vertragsstaaten:

CH, DE, FR, GB, IT, LI, NL, SE

72 Erfinder:

Bryg, William R., Saratoga, CA 95070, US; Gardner,
Michael E., Sunnyvale, CA 94086, US; Boettner,
Steven C., San Jose, CA 95120, US

54 Verfahren zur Aktualisierung der Informationen in einem Adressenübersetzungspufferspeicher.

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patentamt inhaltlich nicht geprüft.

DE 36 89 276 T 2

Verfahren zur Aktualisierung der Informationen in
einem Adressenübersetzungspufferspeicher

Die Erfindung betrifft ein Verfahren zum Aktualisieren von Information in einem Übersetzungspufferspeicher gemäß dem Oberbegriff des Anspruches 1. Solch ein Verfahren ist aus der EP-A-0 115 179 bekannt.

Dieses Verfahren ist auf Computer anwendbar, die Übersetzungspufferspeicher (TLB's) bei der Übersetzung von virtuellen Adressen in physikalische Adressen verwenden. In einer Computerumgebung werden virtuelle Adressen von der Software verwendet, um auf Befehle und Daten zu verweisen, während physikalische Adressen die aktuellen physikalischen Plätzen im Speicher bedeuten, wo die Befehle und Daten gespeichert sind. Die Verwendung beider Arten von Adressen erfordert es, daß eine Übersetzung von virtuellen Adressen in physikalische Adressen stattfindet, sodaß eine Adresse, auf die in der Software verwiesen wird, in einem Zugriff auf die zugehörige physikalische Adresse resultiert.

Im allgemeinen wird der Raum von virtuellen Adressen viel größer sein als der Raum von physikalischen Adressen. Der virtuelle und der physikalische Adreßraum sind typischerweise in Speicherblöcke gleicher Abmessungen, sogenannte Seiten, geteilt, so daß die Übersetzung von einer virtuellen Adresse in eine physikalische Adresse eine Übersetzung von virtuellen Seitenzahlen in physikalische Seitenzahlen mit sich bringt. Ein Seitenverzeichnis (PDIR) liefert die Übersetzung zwischen virtuellen Adressen und physikalischen Adressen. Das Seitenverzeichnis enthält eine Eintragung für jede physikalische Seitenzahl, die mit einer virtuellen Sei-

tenzahl verknüpft ist. Daher ist die direkte Benutzung des Seitenverzeichnisses typischerweise zu langsam, um die Übersetzung durchzuführen. Um die Geschwindigkeit der Übersetzung zu erhöhen, verwenden viele Computer einen Cache-Speicher, der als ein Übersetzungspufferspeicher (TLB) bezeichnet wird, um bei der Übersetzung von virtuellen Adressen in physikalische Adressen zu helfen.

Ein Vorteil eines Cache-Speichers ist, daß ein Speicherzugriff auf den Cache-Speicher typischerweise viel schneller erfolgt als ein Speicherzugriff auf den Hauptspeicher. Typischerweise erfordert diese erhöhte Zugriffsgeschwindigkeit, daß der Cache-Speicher klein ist. In vielen Fällen kann der TLB nicht das vollständige Seitenverzeichnis enthalten, so daß Prozeduren durchgeführt werden müssen, um den TLB zu aktualisieren. Wenn auf eine virtuelle Seite zugegriffen wird, die nicht im TLB vorhanden ist, wird auf das Seitenverzeichnis zugegriffen, um die Übersetzung dieser virtuellen Seitenzahl in eine physikalische Seitenzahl zu bestimmen, und diese Information wird in den TLB eingetragen. Ein Zugriff auf eine Seitentabelle kann größenordnungsmäßig fünfmal länger dauern als ein Zugriff auf den TLB, und daher wird die Geschwindigkeit einer Programmausführung optimiert, indem die in dem TLB verwendeten Übersetzungen aktiv gehalten werden.

Bei vielen Systemen besteht der physikalische Speicher aus einem Hilfsspeicher (wie beispielsweise einem Diskettenspeicher oder einem Magnetbandspeicher), einem Hauptspeicher und einem Cache-Speicher. Der Hilfsspeicher ist typischerweise größer als der Hauptspeicher, wodurch größere Programme durchgeführt werden können, als wenn nur der Hauptspeicher verfügbar ist. In Abhängigkeit von der Länge eines Programms und von der Konkurrenz mit anderen Programmen um den Haupt-

speicher wird ein Programm teilweise oder vollständig in den Hauptspeicher zu einer gegebenen Zeit geladen. Um die Geschwindigkeit des Speicherzugriffs zu erhöhen, verwenden viele Computer einen zusätzlichen Speicher, einen sogenannten Cache-Speicher. Ein Cache-Speicher ist typischerweise schneller als der Hauptspeicher, aber um solche erhöhten Geschwindigkeiten zu erreichen, ist der Cache-Speicher typischerweise viel kleiner als der Hauptspeicher. Daher kann in einer typischen Situation nur ein Teil des Programmsegmentes im Hauptspeicher vom Hauptspeicher in den Cache-Speicher geladen werden. Die Geschwindigkeit der Programmausführung wird optimiert, indem der TLB mit den Übersetzungen für solche virtuellen Adressen, die im Augenblick aktiv verwendet werden, geladen bleibt. Cache-Speicher beruhen auf der Annahme, daß aufgrund der Bezugnahme auf einen speziellen Speicherplatz auf diesen Platz und auf die sehr nahe an diesem liegenden Plätze in naher Zukunft sehr wahrscheinlich zugegriffen werden wird. Diese Eigenschaft wird als Örtlichkeit bezeichnet. Daher wird der Cache-Speicher Daten enthalten, auf die gerade Bezug genommen wurde, und der TLB wird Übersetzungen enthalten, die mit solchen Seiten verknüpft sind.

In Figur 1 ist die Antwort auf das Auftreten einer virtuellen Adresse während der Programmausführung dargestellt. Wenn die Übersetzung für diese virtuelle Adresse in dem TLB vorhanden ist (was als ein TLB-Treffer bezeichnet wird), so wird dann die zugehörige physikalische Adresse von dem TLB hergeleitet und dazu verwendet, um auf den physikalischen Speicher zuzugreifen. Wenn die Übersetzung für diese virtuelle Adresse in dem TLB nicht vorhanden ist (was als TLB-Fehlschlag bezeichnet wird), so wird die Übersetzung für diese virtuelle Adresse in dem Seitenverzeichnis gesucht. Wenn die Übersetzung in dem Seitenverzeichnis vorhanden ist, wird diese Information dann in den TLB eingefügt, und die

virtuelle Adresse ist wieder vorhanden. Diese Zeit eines TLB-Treffers ist, wie wir versichern, derart, daß die resultierende physikalische Adresse zum Zugriff auf den physikalischen Speicher benutzt wird.

Wenn die virtuelle Adresse auf einer Seite von virtuellen Adressen vorhanden ist, mit der keine Seite von physikalischen Adressen verknüpft ist, dann wird es für diese Seite keinen Eintrag in dem Seitenverzeichnis geben. Solch ein Ereignis wird als ein Seitenfehler bezeichnet. Als Antwort auf einen Seitenfehler wird die virtuelle Seite, auf die Bezug genommen wird, einer physikalischen Seite zugewiesen und diese Information in das Seitenverzeichnis eingefügt. Wenn alle physikalischen Seiten bereits mit anderen virtuellen Seiten verknüpft sind, muß das Seitenfehlersteuerprogramm auswählen, welche von den physikalischen Seiten der virtuellen Adresse, auf die im Augenblick Bezug genommen wird, wieder zugewiesen wird. Es gibt viele Algorithmen für eine solche Wahl, einschließlich (zuerst-rein-zuerst-raus)- und (vor kurzem am wenigsten verwendet)-Algorithmen. Weil dieser Zugriffsvorgang komplizierter als solche Routinen sind, die typischerweise im Mikrocode durchgeführt werden, wird das Seitenfehlersteuerprogramm typischerweise in der Software durchgeführt. Mikrocode ist typischerweise einfach gehalten, sodaß er in relativ kleine und schnelle Speicher passen kann, die zum Speichern des Mikrocodes verwendet werden.

Nach Beendigung der Seitenfehlersteuerprogrammroutine ist die virtuelle Adresse wieder vorhanden. Der TLB hat noch keine Übersetzung, sodaß es einen TLB-Fehlschlag geben wird. Daher wird die Übersetzung in dem Seitenverzeichnis nachgeschlagen, die Übersetzung in den TLB eingefügt, und die virtuelle Adresse ist zum dritten Mal vorhanden. Diesmal wird ein

TLB-Treffer sichergestellt, sodaß die resultierende physikalische Adresse zum Zugriff auf den physikalischen Speicher verwendet wird.

Die obige Prozedur hat den Nachteil, daß sie zweimal ein Zugreifen auf das Seitenverzeichnis erfordert. Dies ist von Nachteil, weil das Nachschlagen einer Übersetzung in ein Seitenverzeichnis größenordnungsmäßig fünfmal länger dauern kann als das Hochsetzen einer Übersetzung in den TLB.

Der in der EP-A-0 115 179 beschriebene Stand der Technik umfaßt Software-Mittel, welche die Inhalte der TLB's mit den Inhalten der Seitentabelle in dem Hauptspeicher synchronisieren, wenn Änderungen in der Eintragung der virtuellen-realen Adresse gemacht werden. Eintragungen sowohl in den TLB's als auch in den Seitenrahmentabellen müssen ungültig werden, um sicherzustellen, daß eine überholte Eintragungsinformation nicht für eine nachfolgende Übersetzung verwendet wird.

Das bekannte Verfahren stellt drei Funktionen bereit, um bei der Synchronisation der TLB-Eintragungen mit den Inhalten der Seitentabelle in dem Hauptspeicher zu helfen. Diese Funktionen können verwendet werden, um die gesamten TLB-Inhalte für ungültig zu erklären oder nur ausgewählte TLB-Eintragungen für ungültig zu erklären. Diese Funktionen werden von I/O-Schreibbefehlen aufgerufen, die auf spezifische I/O-Adressen in dem Block der von dem System erkannten I/O-Adressen gerichtet sind. Spezifische Bits der von einem I/O-Schreibbefehl übertragenen Daten werden verwendet, um einen Segmentanzeiger auszuwählen. Alle TLB-Eintragungen, die diesen Segmentanzeiger aufweisen, werden ungültig gesetzt. Nachfolgende Übersetzungen mit einer effektiven Adresse in dem ungültig gesetzten Segment werden veranlassen, daß die TLB-Inhalte aus der Seitentabelle in dem Hauptspeicher ak-

tualisiert werden.

Da eine weitere Geschwindigkeitserhöhung des Übersetzungsvorganges erwünscht ist, ist es eine Aufgabe der Erfindung, sicherzustellen, daß es nach Beendigung einer Seitenfehlersteuerprogrammroutine einen TLB-Treffer statt eines TLB-Fehlschlages geben wird.

Diese Aufgabe wird erfindungsgemäß durch die kennzeichnenden Merkmale des Anspruchs 1 gelöst.

Der abhängige Anspruch 2 kennzeichnet eine vorteilhafte Weiterentwicklung.

Der Satz der eingeführten Software-Befehle erlaubt es der Software, die Übersetzung direkt in den TLB einzufügen. Dies versetzt das Seitenfehlersteuerprogramm in die Lage, die Übersetzung nicht nur in das Seitenverzeichnis einzufügen, sondern auch die Information in den TLB einzufügen. Daher wird nach Beendigung der Seitenfehlersteuerprogrammroutine statt der Sicherstellung, daß es einen TLB-Fehlschlag geben wird, sichergestellt, daß einen TLB-Treffer geben wird.

➤ In vielen Ausführformen von Computern, die TLB's und Cache-Speicher verwenden, gibt es einen Cache für Befehle und einen getrennten Cache für Daten. Zusätzlich gibt es getrennte TLB's für den Befehlecache und für den Datencache. Bei solchen Ausführungsformen sind getrennte Software-Befehle vorhanden, die es ermöglichen, daß die Übersetzungen direkt in den Daten-TLB und in den Befehle-TLB eingefügt werden. Bei einigen Ausführungsformen ist es auf ähnliche Weise für jede dieser Kapazitäten vorteilhaft, von einem Befehlspaar ausgeführt zu werden. Bei einer speziellen Ausführungsform fügt der erste Befehl dieses Befehlspaares die Übersetzung zw-

schen die virtuelle Seitenzahl und ihre zugehörige physikalischen Seitenzahl ein. Der zweite Befehl dieses Befehlpaars fügt irgend eine Schutzinformation, Flags oder andere Information ein, die zur Überprüfung der Legalität des Zugriffs auf die Seite verwendet werden können.

Die verschiedenen Merkmale und Vorteile der vorliegenden Erfindung können leichter unter Bezug auf die folgende detaillierte Beschreibung in Verbindung mit den begleitenden Zeichnungen verstanden werden.

Figur 1 zeigt eine typische Prozedur, um auf virtuelle Adressen zu antworten, die während der Ausführung eines Programms auftreten.

Figur 2 stellt den Zugriff auf einen Übersetzungspufferspeicher dar.

Figur 3 stellt den Zugriff auf einen Cache-Speicher dar.

Figur 4 stellt den Zugriff auf ein typisches Seitenverzeichnis dar.

Figur 5 stellt die Prozedur dar, die Software-Befehle verwendet, welche eine Einfügung von Information direkt in den TLB ermöglicht, um auf virtuelle Adressen zu antworten, die während der Ausführung eines Programms auftreten.

In Figur 2 ist ein Zugriffsverfahren auf Eintragungen in einem Übersetzungspufferspeicher (TLB) als Antwort auf das Auftreten einer virtuellen Adresse gezeigt. Die virtuelle Adresse wird in ein Register 21 geladen. Die niedrigeren L Bits der virtuellen Adresse sind der physikalische Abzug und bezeichnen die Zeilenzahl einer Eintragung in einer vir-

tuellen Seite mit 2^L Zeilen. Die verbleibenden Bits im Register bezeichnen die virtuelle Seitenzahl. Bei einer Ausführungsform, in der ein TLB 2^N Eintragungen und der virtuelle Adressraum 2^{M+N} Seiten enthält, ist dann ein Algorithmus erforderlich, um die virtuelle Seitenzahl in eine als Index bezeichnete N-Bit-Zahl umzuwandeln. In einer in Figur 2 gezeigten speziellen Ausführungsform wird dies erreicht, indem die geringsten signifikanten N Bits einer virtuellen Seitenzahl verwendet werden. Die verbleibenden M Bits der virtuellen Seitenzahl werden als virtuelle Markierung bezeichnet.

Der Index wird als eine Adresse verwendet, um auf den TLB zuzugreifen. Ein Komparator 23 vergleicht die virtuelle Markierung aus Register 21 mit den M Bits des TLB, um zu prüfen, ob es eine Übereinstimmung gibt. In dieser speziellen Ausführungsform bilden die M signifikantesten Bits des TLB die TLB-Markierung. Wenn es eine Übereinstimmung gibt, ist der Ausgang des Komparators 23 wahr, was anzeigt, daß es einen TLB-Treffer gegeben hat. Andernfalls ist der Ausgang des Komparators falsch, was anzeigt, daß es einen TLB-Fehlschlag gegeben hat. Bei einer Ausführungsform mit 2^P Seiten des physikalischen Speichers zeigen P der TLB-Bits jeder TLB-Eintragungen die physikalische Adresse für diese Eintragung an. Wenn es einen Treffer gibt, werden diese Bits als physikalische Seitenzahl verwendet, die mit der virtuellen Adresse im Register 21 verknüpft ist. Die physikalische Adresse wird mit dieser physikalischen Seitenzahl verknüpft, um die physikalische Adresse zu erzeugen. Typischerweise enthält jede TLB-Eintragung zusätzlich zu den in Figur 2 ausdrücklich gezeigten M+P Bits einen Satz von Bits, um andere Information, beispielsweise Schutzinformation oder Flags, für jede physikalische Seite zu halten.

In Figur 3 ist gezeigt, wie eine physikalische Adresse ver-

wendet wird, um auf einen Cache-Speicher 32 zuzugreifen. Jede Eintragung im Cache-Speicher 32 enthält ein M-Bit-Cache-Ende und ein 32-Bit-Segment der Cache-Daten. Welche Eintragung des Cache 32 ausgewählt wird, wird von der in ein Register 31 geladenen physikalischen Adresse bestimmt. In dieser speziellen Ausführung wird ein 32-Bit-Speicher verwendet, sodaß jede Eintragung vier Datenworten enthält. Um unter diesen Worten auszuwählen, werden die beiden niedrigsten Bits der physikalischen Adresse als Eingabe an einen Multiplexer 34 verwendet. Die nächsten N geringsten signifikanten Bits werden als Cache-Speicherindex bezeichnet und werden verwendet, um die Adresse, auf die in dem Cache-Speicher zugegriffen wird, auszuwählen. Ein Komparator 33 vergleicht die M signifikantesten Bits des Registers 31 mit den M signifikantesten Bits der Cache-Speichereintragung bei der von dem Cache-Speicherindex angezeigten Adresse. Der Ausgang des Komparators 33 ist wahr, was anzeigt, daß es einen Cache-Speichertreffer gegeben hat, wenn die Markierung der physikalischen Adresse dasselbe ist wie bei der Eintragung, auf die im Cache-Speicher zugegriffen wird. Andernfalls ist der Ausgang des Komparators 33 falsch, was anzeigt, daß es einen Cache-Fehlschlag gegeben hat.

In Fig. 4 ist eine Ausführungsform eines Seitenverzeichnis dargestellt. Um zu vermeiden, nacheinander jede Eintragung in dem Seitenverzeichnis durchsehen zu müssen, wenn das Seitenverzeichnis eine der gegebenen virtuellen Adresse entsprechende Eintragung enthält, ist es von Vorteil, die Eintragungen in dem Seitenverzeichnis in Sätze zu teilen. Ein Zerhackalgorithmus kann verwendet werden, um eine virtuelle Adresse in einen Zeiger umzuwandeln, der auf die erste Eintragung in dem Satz zeigt, zu welchem die virtuelle Adresse gehört. In Fig. 4 werden die verschiedenen Sätze durch die Buchstaben A, B, ... Z angezeigt. Die Eintragungen in jedem

Satz bilden eine Verbindungsliste, so daß nur die Eintragungen in ihrem zugehörigen Satz nacheinander überprüft zu werden brauchen, anstatt das Eintragungsseitenverzeichnis durchsuchen zu müssen, ob eine gegebene virtuelle Seitenzahl in das Seitenverzeichnis eingetragen wurde. Wenn alle Eintragungen in dem zugehörigen Satz überprüft wurden und keine mit der gesuchten virtuellen Adresse übereinstimmt, zeigt dies dann an, daß es einen Seitenfehler gibt.

Fig. 5 zeigt ein Flußdiagramm einer Prozedur, die die in den Figuren 2 bis 4 gezeigten Komponenten verwendet, um auf das Auftreten einer virtuellen Adresse während einer Programmausführung zu antworten. In Schritt 51 ist die virtuelle Adresse dadurch vorhanden, daß die virtuelle Adresse in das Register 21 eingefügt wird. Wenn die Übersetzung für diese virtuelle Adresse in dem TLB vorhanden ist (was als TLB-Treffer bezeichnet wird), wird die zugehörige physikalische Adresse von dem TLB hergeleitet und in Schritt 52 zum Zugriff auf den physikalischen Speicher verwendet. Wenn die Übersetzung für diese virtuelle Adresse in dem TLB nicht vorhanden ist (was als TLB-Fehlschlag bezeichnet wird), dann wird in Schritt 53 die Übersetzung für diese virtuelle Adresse in dem Seitenverzeichnis gesucht. Wenn die Übersetzung in dem Seitenverzeichnis vorhanden ist, dann wird in Schritt 54 diese Information in den TLB eingefügt, und die virtuelle Adresse ist wieder vorhanden. Diese Zeit eines TLB-Treffers ist, wie wir versichern, derart, daß die resultierende physikalische Adresse zum Zugriff auf den physikalischen Speicher benutzt wird.

Wenn die virtuelle Adresse auf einer Seite der virtuellen Adressen vorhanden ist, mit der keine Seite von physikalischen Adressen verknüpft ist, dann wird es keine Eintragung für diese Seite in das Seitenverzeichnis geben. Solch ein

Ereignis wird als Seitenfehler bezeichnet. Wenn es einen Seitenfehler gibt, dann wird in Schritt 55 die virtuelle Seite, auf die Bezug genommen wird, einer physikalischen Seite zugeordnet und diese Information in das Seitenverzeichnis eingefügt. Wenn alle physikalischen Seiten bereits mit anderen virtuellen Seiten verknüpft sind, muß das Seitenfehlersteuerprogramm auswählen, welche physikalische Seite der virtuelle Adreßseite, auf die gerade Bezug genommen wird, erneut zugeordnet werden soll. Es gibt viele Algorithmen für eine solche Wahl, einschließlich (zuerst-rein-zuerst-raus)- und (vor kurzem am wenigsten verwendet)-Algorithmen. Da dieser gesamte Vorgang komplizierter als solche typischerweise im Mikrocode ausgeführte Routinen sind, wird das Seitenfehlersteuerprogramm typischerweise in der Software durchgeführt.

Die Software umfaßt explizite Befehle, die Information in den TLB einfügen. Wenn ein solcher Befehl ausgeführt wird, ergibt sich das direkte und definierte Ergebnis, daß die Information in den TLB eingefügt wird. Im Gegensatz dazu wird bei früheren Verfahren zum Aktualisieren des TLB, wie oben in Bezug auf Fig. 1 beschrieben wurde, Information in den TLB nur als ein indirektes Ergebnis von gewissen Software-Befehlen eingefügt. Wenn beispielsweise eine virtuelle Adresse auftritt, kann die Information in Abhängigkeit, ob es einen TLB-Treffer oder einen TLB-Fehlschlag gibt, in dem TLB geändert werden oder nicht. Folglich gibt es kein definiertes Resultat an dem TBL wie ein Resultat des Auftretens der virtuellen Adresse. Eine explizite Kontrolle über die Eintragung von Information in den TLB liefert einen viel flexibleren und wirkungsvolleren Weg, die Inhalte des TLB zu kontrollieren. Am Ende einer Seitenfehlersteuerprogrammroutine ermöglichen es die expliziten Software-Befehle zum Einfügen von Daten in den TLB zusätzlich zum Eintragen der

Übersetzungsinformation in das Seitenverzeichnis, daß diese Information auch in den TLB eingefügt wird. Dies führt zu einem wirkungsvolleren Verfahren zum Aktualisieren der Information im TLB. Im Flußdiagramm der Fig. 5 ist höchstensfalls ein Zugriff auf das Seitenverzeichnis erforderlich, wohingegen in dem Flußdiagramm der Fig. 1 zwei Zugriffe auf das Seitenverzeichnis erforderlich sind, wenn es einen Seitenfehler gibt. Zusätzlich erlaubt diese Zusatzkontrolle über den TLB, die TBL-Inhalte in einer flexibleren Weise zu verändern. Mehrfacheintragungen können unter Software-Kontrolle durchgeführt werden. Auch können TBL-Eintragungen vorgeladen werden, um zu garantieren, daß TBL-Fehler nicht während kritischen Codeabschnitten auftreten, oder um die Geschwindigkeit des Vorgangs zu erhöhen, indem die Zahl von TLB-Fehlschlägen verringert wird.

In einigen Ausführungen wird es mehr Informationsbits in jeder TLB-Eintragung geben, als in Antwort auf einen einzelnen Software-Befehl eingetragen werden können. Beispielsweise können in der bevorzugten Ausführung nur bis zu 32 Bits von Daten in einem einzelnen Befehlszyklus bewegt werden, und alle Befehle sind für den Betrieb in einem einzelnen Zyklus vorgesehen. Jedoch hat jede TBL-Eintragung mehr als 32 Bits. Daher ist mehr als ein explizierter Softwarebefehl erforderlich, um alle Information einzutragen.

In jede TLB-Eintragung ist ein Bit eingeschlossen, das anzeigt, ob diese Eintragung gültig ist. Wann immer dieses Bit gesetzt ist, wird die TLB-Markierung für diese Eintragung von dem Komparator 23 nicht gefunden, um die virtuelle Markierung irgend einer virtuellen Adresse anzupassen. Dies verhindert, daß die Information verwendet wird. Sondern eine solche ungültige Information wird entweder eventuell als gültig markiert werden, wenn alle Information eingegeben

sind, oder wird eventuell als Teil einer nachfolgenden Aktualisierung der Inhalte des TLB wieder ersetzt. In einer besonderen Ausführung werden als Antwort auf den ersten expliziten Befehl die virtuelle Seitenzahl und die physikalische Seitenzahl in den TLB eingefügt. Zusätzlich setzt dieser Befehl diese Eintragung als ungültig. Dies wird so ausgeführt, daß, wenn die Aktualisierung an dieser Stelle unterbrochen wird (beispielsweise durch einen Maschinenfehler oder von einem externen Unterbrechung, wie er in einer zeitgeteilten Umgebung auftritt), dann diese teilweisen Daten nicht verwendet werden können und keine Fehler erzeugen können. Wenn alle Information eingetragen ist, wird diese TLB-Eintragung als gültig markiert. Bei dem zweiten expliziten Befehl wird eine Schutzinformation eingefügt, die es erlaubt, Seiten vor einer unautorisierten Eintragung zu schützen. Der zweite Befehl präsentiert sowohl die virtuelle Adresse als auch die Schutzinformation, sodaß die virtuelle Adresse gegenüber derjenigen virtuellen Adresse überprüft werden kann, die von dem ersten Befehl eingefügt wurde, um so sicherzustellen, daß keine Falscheinfügung von Daten auftritt.

Ansprüche

1. Verfahren zur Aktualisierung der Informationen in einem Übersetzungspufferspeicher (TLB), wobei das Verfahren die Schritte aufweist:
 - (a) Zuordnen einer physikalischen Speicherseitennummer zu einer virtuellen Seitennummer in einem Seitenverzeichnis, um eine virtuelle Adresse (51) in ihre entsprechende physikalische Adresse (52) zu übertragen, wobei der TLB einen Teil der Eintragungen des Seitenverzeichnisses aufnimmt (54), um die Geschwindigkeit der Übersetzung zu erhöhen; und
 - (b) als Reaktion auf einen Seitenfehler (53) in dem Seitenverzeichnis Verweisen auf die fehlerhafte Seite und Aktualisierung (55) einer Eintragung in dem Seitenverzeichnis durch ein Seitenfehlersteuerprogramm, dadurch gekennzeichnet, daß gleichzeitig mit der Aktualisierung in dem Seitenverzeichnis die entsprechende Information durch eine explizite Anweisung in den Übersetzungspufferspeicher eingefügt wird.
2. Verfahren nach Anspruch 1, bei dem eine Einfügung von Information in den TLB mehr als einen expliziten Software-Befehl erfordert, bei dem zusätzlich zu der Einfügung von Information in eine TLB-Eintragung durch einen ersten der expliziten Software-Befehle die Eintragung als ungültig markiert wird; und bei dem die Eintragung als gültig markiert wird, wenn alle Information eingetragen worden ist.

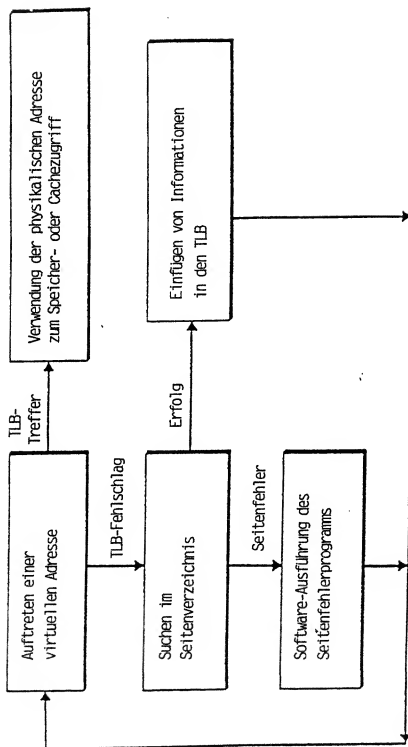


FIG 1

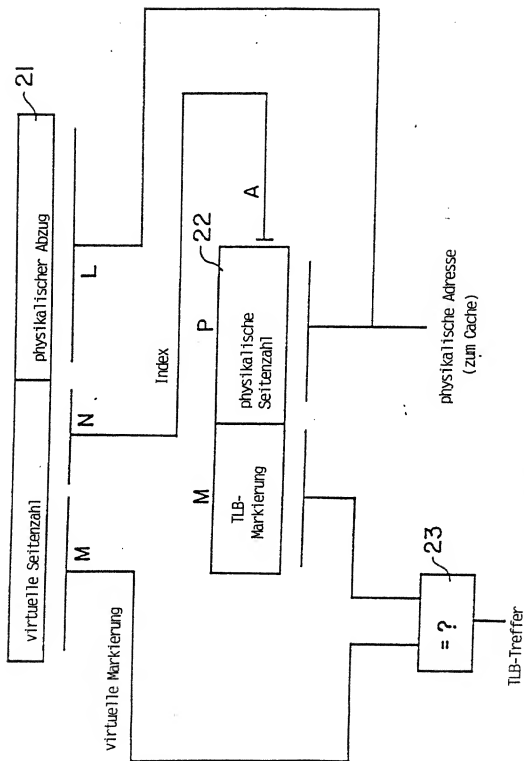


FIG 2

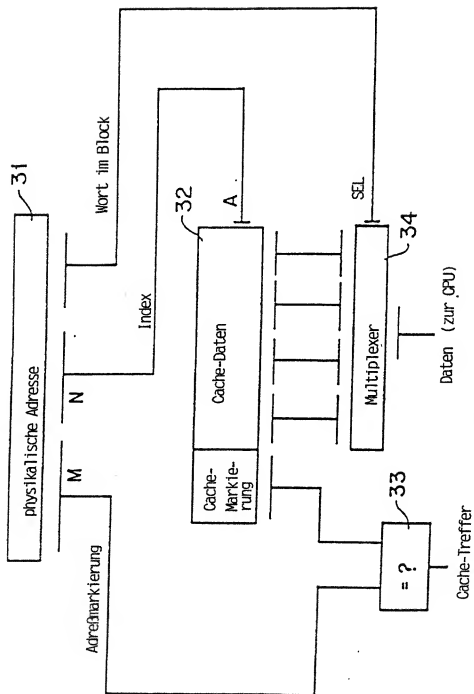


FIG 3

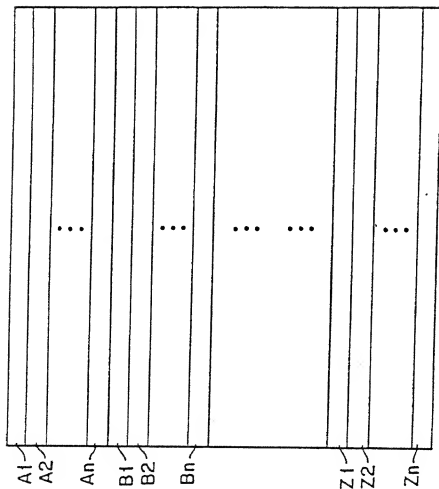


FIG 4

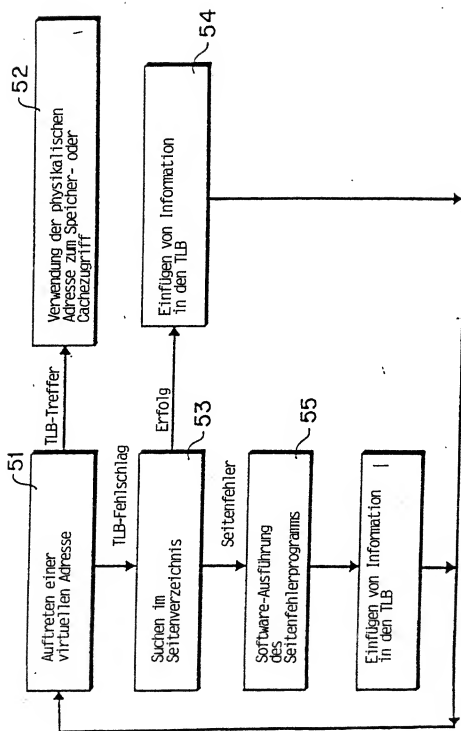


FIG 5